

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra telekomunikační techniky

# **Absolvování individuální odborné praxe**

## **Individual Professional Practice in the Company**

# Zadání bakalářské práce

Student:

**Daniel Walter**

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R059 Mobilní technologie

Téma:

Absolvování individuální odborné praxe  
Individual Professional Practice in the Company

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: Stora Enso Wood Products Zdirec s.r.o.
2. Struktura závěrečné zprávy:
  - a. Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
  - b. Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
  - c. Zvolený postup řešení zadaných úkolů.
  - d. Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
  - e. Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
  - f. Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vedl odbornou praxi studenta

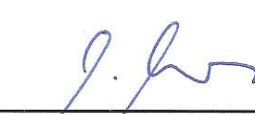
Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Zdeňka Chmelíková, Ph.D.**

Konzultant bakalářské práce: Ing. Ivo Pajurek

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018

  
doc. Ing. Miroslav Vozňák, Ph.D.  
vedoucí katedry



  
prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární  
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 27. dubna 2018

.....*Melny*.....

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 27. dubna 2018

..... 

Stora Enso Wood Products Ždírec s. r. o.  
582 63 Ždírec n. Doubravou, Nádražní 66

Rád bych na tomto místě poděkoval Ing. Ivo Pajurkovi za vedení a odbornou konzultaci při výkonu mé odborné praxe, a také celému SAP Basis teamu ze společnosti Stora Enso za věcné rady a poskytnuté odborné materiály.

## **Abstrakt**

V této bakalářské práci popisuji průběh individuální odborné bakalářské praxe ve společnosti Stora Enso Wood Products Ždírec, s.r.o. Práce obsahuje informace o odborném zaměření firmy, zadáných úkolech, jejich vyhotovení a popis technologií, které byly použity při řešení. Závěrečná část popisuje získané vědomosti a celkově hodnotí přínos absolvované praxe.

**Klíčová slova:** SAP, ABAP, Fiori, Open UI5, odborná praxe, Stora Enso

## **Abstract**

In this Bachelor thesis, I describe process of individual professional practice in the company Stora Enso Wood Products Ždírec, s.r.o. The bachelor thesis contains information about specialization of the company, my tasks, their implementation and technologies used in solution. The last part describes acquired knowledge and skills and overall assessment of benefits gained by this professional practice.

**Key Words:** SAP, ABAP, Fiori, Open UI5, professional practice, Stora Enso

# Obsah

<b>Seznam použitých zkratk a symbolů</b>	<b>8</b>
<b>Seznam obrázků</b>	<b>9</b>
<b>Seznam výpisů zdrojového kódu</b>	<b>10</b>
<b>1 Úvod</b>	<b>11</b>
1.1 Odborné zaměření společnosti . . . . .	11
1.2 Popis pracovní pozice . . . . .	11
<b>2 Seznam zadaných úkolů a vyjádření jejich časové náročnosti</b>	<b>13</b>
2.1 Aplikace nebo funkční moduly pro SAP . . . . .	13
2.2 Dokumentace rolí . . . . .	13
2.3 Mobilní vývoj . . . . .	13
<b>3 Použité technologie a nástroje</b>	<b>14</b>
3.1 SAP GUI, ABAP . . . . .	14
3.2 Vývoj mobilních aplikací . . . . .	15
<b>4 Řešení zadaných úkolů</b>	<b>17</b>
4.1 Aplikace nebo funkční moduly pro SAP . . . . .	17
4.2 Mobilní vývoj . . . . .	28
4.3 Aplikace pro schvalování faktur na platformě SAP Neptune . . . . .	30
<b>5 Uplatnění teoretických a praktických znalosti a dovednosti</b>	<b>36</b>
<b>6 Scházející znalosti a dovednost</b>	<b>37</b>
<b>7 Dosažené výsledky</b>	<b>38</b>
<b>8 Závěr</b>	<b>39</b>
<b>Literatura</b>	<b>40</b>

## Seznam použitých zkratek a symbolů

Idoc	– intermediate document
GUI	– Graphical user interface
S4HANA	– SAP Business Suite 4 SAP HANA
HTML	– Hypertext Markup Language
CSS	– Cascading Style Sheets
PHP	– Hypertext Preprocessor
OData	– Open Data Protokol
XML	– eXtensible Markup Language
JSON	– JavaScript Object Notation
REST	– Representational state transfer
IDE	– Integrated Development Environment
VPN	– Virtual private network
RFC	– Remote Function Call



## Seznam obrázků

1	Třívrstvá architektura klient-server[2] . . . . .	14
2	SAP Fiori On-Premis vs SAP Fiori Cloud [3] . . . . .	16
3	Migrace repositáře - vlastní zpracování . . . . .	18
4	Ukázka úvodní obrazovky programu pro migraci repositářů - vlastní zpracování .	18
5	Zobrazení logu v transakci SLG1 - vlastní zpracování . . . . .	19
6	Ukázka volání funkce pro zjištění stavu dokumentu - vlastní zpracování . . . . .	20
7	Transakční kód SE91 pro správu a předpřipravených zpráv - vlastní zpracování .	20
8	Tabulka pro správu funkcí po přihlášení uživatele v transakčním kódu SE16 - vlastní zpracování . . . . .	21
9	Ukázka uložených informací o uživateli WALTEDA01 - vlastní zpracování . . . . .	22
10	Ukázka transportu - vlastní zpracování . . . . .	23
11	Ukázka výstupu programu - vlastní zpracování . . . . .	24
12	Vytvoření nového ticketu do systému Service Now - vlastní zpracování . . . . .	25
13	Dialog pro zasílání ticketu do nového systému ServiceNow - vlastní zpracování . .	25
14	Zobrazení dostupných technických rolí pro kontrolora - vlastní zpracování . . . .	27
15	Ukázka přidávání nového záznamu s možnostmi modulů - vlastní zpracování . . .	28
16	Ukázka SAP Fiori Launchpadu - vlastní zpracování . . . . .	28
17	Webové prostředí WebIDE pro vývoj na platformě SAP Fiori - vlastní zpracování	30
18	Vývojové prostřední na platformě Neptune - vlastní zpracování . . . . .	31
19	Implementace třídy pro komunikaci s platformou - vlastní zpracování . . . . .	31
20	Implementace třídy pro komunikaci s platformou - vlastní zpracování . . . . .	32
21	Ukázka vývoje aplikace v platformě NEPTUNE - vlastní zpracování . . . . .	33
22	Ukázka funkce pro zobrazení stránky s podrobnými informacemi o dokumentu - vlastní zpracování . . . . .	34
23	Původní aplikace v SAP GUI (transakční kód ME52N) - vlastní zpracování . . .	35
24	Výsledná aplikace pro schvalování faktur na platformě Neptune - vlastní zpracování	35

## Seznam výpisů zdrojového kódu

1	Ukázka volání Authority checku . . . . .	26
2	Ukázka OData feedu z webové služby . . . . .	29
3	Ukázka funkce v programovacím jazyce ABAP . . . . .	32

# 1 Úvod

Bakalářskou práci formou absolvování odborné praxe jsem si zvolil, jelikož pracuji ve společnosti Stora Enso na studentské pozici a zabývám se zde programováním v SAP systému. O této možnosti jsem se dozvěděl na konci druhého ročníku při zjišťování bližších informací prostřednictvím webových stránek univerzity. Jelikož se společnost Stora Enso nachází na seznamu školou akceptovaných subjektů, bylo mi mým konzultantem vyhověno a dohodli jsme se na úkolech, které budu dále popisovat. Cílem této práce bylo vytvoření několika programů, které pomáhají ulehčit každodenní práci s informačním systémem SAP nebo přispívají k její automatizaci.

Ve společnosti Stora Enso Wood Products Ždírec, s. r. o. jsem pracoval už od druhého ročníku na pozici se zaměřením na vývoj mobilních aplikací pro ERP SAP, s kterým jsem dříve neměl žádné předchozí zkušenosti. Během třetího semestru probíhala migrace SAP systému do nového datacentra, kde jsem se podílel na vývoji podpůrných aplikací pro sběr a analýzu uživatelských dat, které později pomohly k správnému nastavení uživatelských SAP GUI klientů. Během odborné praxe jsem měl možnost spolupracovat i s jinými týmy ve firmě, a tím jsem získal další zkušenosti i v jiných odvětvích.

V této bakalářské práci se budu nejdříve zabývat popisem odborného zaměření firmy, zadanými úkoly, na kterých jsem pracoval v průběhu dvou semestrů, ale také se zaměřím na teoretické znalosti, nové technologie, se kterými jsem měl možnost seznámit se a které byly v praxi použity.

## 1.1 Odborné zaměření společnosti

Stora Enso se zaměřuje na prodej a zpracovávání obnovitelných materiálů, hlavně dřeva. Portfolio produktů a služeb je velmi rozsáhlé od jednoduchých výrobků až po celé budovy ze dřeva, bioenergie a nanotechnologie, papír a lepenky. Stora Enso a její divize Wood Products je rozšířena po celém světě. Společnost byla založena v roce 1998 spojením finské společnosti Enso Oyj a švédské společnosti Stora Kopparbergs Bergslags Aktiebolag (STORA). V Ostravě bylo v roce 2006 založeno oddělení Software development and competence centrum. V průběhu let se oddělení rozšířila až na dnešních více než 200 zaměstnanců a stala se nejvýznamnějším IT centrem společnosti sídlícím v Nové Karolině. Společnost se každoročně účastní pracovního veletrhu Kariéra Plus pořádaného VŠB-TU Ostrava. Na tomto veletrhu je možnost se přihlásit na workshop pořádaný společností Stora Enso. Tato společnost dále nabízí možnost osobní návštěvy svého sídla a seznámení se s pracovním prostředím této společnosti.[1]

## 1.2 Popis pracovní pozice

Ve firmě jsem byl přijat na pracovní pozici mobilního vývojáře na platformě Netpune, která slouží jako rozšíření pro systém SAP. Toto rozšíření umožňuje uživatelům systému pracovat s mobilními aplikacemi, a proto nejsou jen odkázáni na desktopovou verzi. Postupem času jsem se začal orientovat i na jiné platformy, jako je Fiori, S4Hana i SAP ECC. Zabýval jsem se jak tvorbou

nových aplikací, tak i úpravami již existujících. Byl jsem přiřazen do SAP Basis technical týmu, který se stará o zajištění každodenního provozu systému a řeší mnoho problémových provozních situací. Během celého roku jsem dostával různé úkoly, které se týkaly mobilních aplikací, dále jsem měl možnost pracovat na nových projektech a tvorbě celých programů pro systém SAP. Také jsem měl možnost vytvářet jednodušší programy v C++, případně Javě, které se později využily při migraci k novému poskytovateli infra služeb. Spolupráce s kolegy byla převážně osobní. Získal jsem zkušenosti od kolegů ze zahraničí, a to prostřednictvím e-mailu nebo Skypu. Tato komunikace byla v anglickém jazyce.

## **2 Seznam zadaných úkolů a vyjádření jejich časové náročnosti**

### **2.1 Aplikace nebo funkční moduly pro SAP**

1. Upravení funkce pro zobrazování faktur z archívu.
2. Aplikace pro migraci původního repositáře do nového repositáře.
3. Aplikace pro úpravu špatných odkazů na dokumenty.
4. Efektivní správa funkcí, které se spouštějí po přihlášení uživatele.
5. Aplikace pro audit a monitoring uživatelů.
6. Aplikace pro kontrolu transportu.
7. Aplikace pro zasílání ticketu ze systému.
8. Aplikace pro zasílání idocu podle přednastavených pravidel.

### **2.2 Dokumentace rolí**

1. Aplikace pro dokumentaci rolí.

### **2.3 Mobilní vývoj**

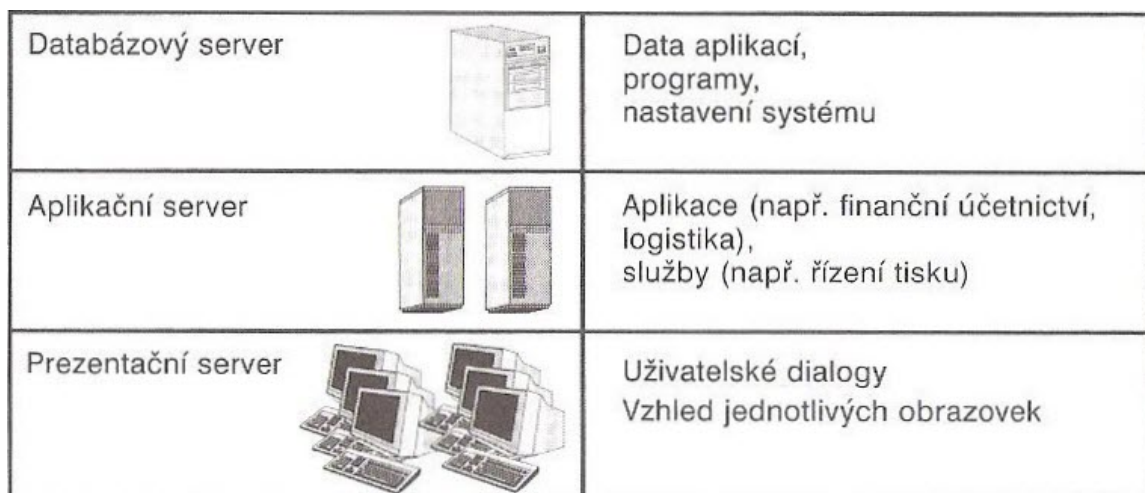
1. Základní princip fungování webových služeb.
2. Ukázka příkladu třídy a prostředí Neptune Designeru, kterou jsem navrhl, jako uživatelský komfort pro použití schvalování faktur.

### 3 Použité technologie a nástroje

Pro vypracování zadaných úkolů jsem nejprve vyhledal technologie a nástroje, které by byly pro řešení vhodné. V některých případech byla technologie předem určena a u dalších úkolů jsem si mohl vybrat a navrhnout své řešení v jazyce, který jsem vyhodnotil jako nejlepší pro daný úkol.

#### 3.1 SAP GUI, ABAP

SAP je softwarový produkt sloužící jako podnikový informační systém, který je dnes nejvíce využíván v korporátní sféře. SAP funguje na principu klient/server a využívá třívrstvý model.[2, 7]



Obrázek 1: Třívrstvá architektura klient-server[2]

Tento systém se skládá z několika tzv. modulů. Moduly slouží k řízení různých firemních procesů. Mezi nejznámější moduly používané ve společnosti Stora Enso patří:

- MM – Materiálové hospodářství (Material Management)
- SD – Prodej a distribuce (Sales and Distribution)
- FI - Finanční účetnictví (Finance)
- CO - Controlling
- HR – Řízení lidských zdrojů (Human Resources)
- PM – Údržba (Plant Maintenance)

**SAP GUI** je jedno z uživatelských rozhraní, které se používá pro grafický přístup do systému SAP. SAP GUI funguje na všech dnes dostupných platformách operačního systému Unix, Mac, Windows.

Pro platformu Windows existuje samostatné SAP GUI určené přímo pro tento operační systém, zatímco pro ostatní platformy Unix a Mac se používá SAP GUI založené na Java platformě. [9]

**ABAP** je programovací jazyk vytvořený pro vývoj aplikací systému SAP. Tento programovací jazyk nabízí klasické strukturované programování a také práci s objekty. Tvůrcem programovacího jazyka ABAP je společnost SAP. [6]

## 3.2 Vývoj mobilních aplikací

### 3.2.1 Javascript

Javascript je skriptovací jazyk, který se používá ve stránkách HTML. Jedná se o jazyk objektový, to znamená, že lze v něm vytvářet objekty a manipulovat s nimi. Právě do objektů je zapouzdřeno okno prohlížeče, obsah stránky a další prvky. Jeho syntaxe vychází z jazyka C++, proto je citlivý na velká a malá písmena.

Navíc je to jazyk beztypový - při deklaraci proměnných se nespecifikuje typ, a konverze hodnot probíhají automaticky. Z tohoto důvodu občas dochází k tomu, že výsledný typ operace je odlišný od plánovaného záměru. Díky své oblibě a jednoduchosti implementace se JavaScript v poslední době stává skriptovacím jazykem nejen na webových stránkách, ale i v aplikacích. Existují i jiné skriptovací jazyky, jako je například VBScript. Ten je ale velmi málo používán. [4]

### 3.2.2 SAPUI5

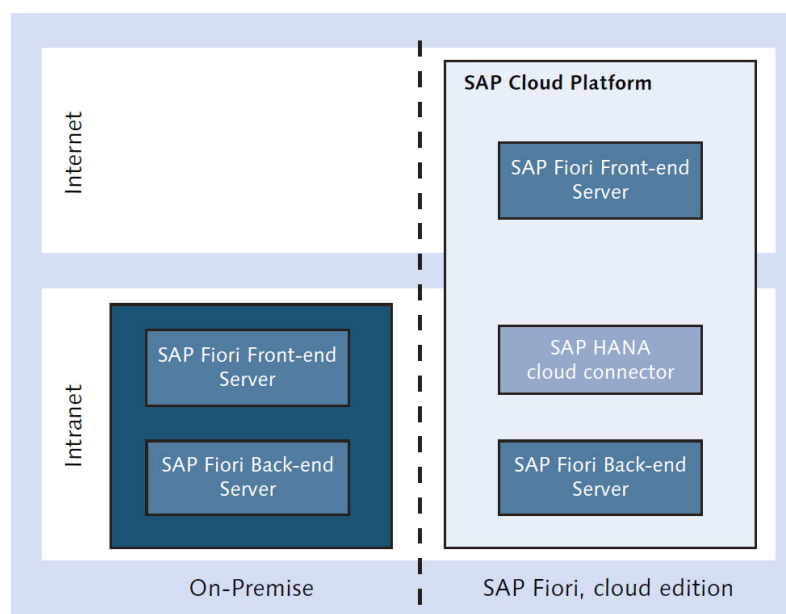
Jedná se o framework, který obsahuje několik knihoven a slouží vývojáři pro vývoj webových a mobilních aplikací. SAPUI5 je založený na OpenUI5, který však neobsahuje všechny tyto knihovny a má jiné licenční podmínky pro použití. SAPUI5 patří pod licenci SAP a integrován do SAP produktů jako je SAP HANA, SAP Cloud Platform apod. S tímto frameworkem můžeme vyvíjet SAP web aplikace pomocí HTML5, CSS a Javascriptu. [8]

### 3.2.3 Neptune a SAP Fiori[5]

Jedná se o dvě různé platformy, které využívají zmíněný SAPUI5 framework. Neptune je produkt, který je certifikovaný SAP a nabízí hlavně využití pro mobilní vývoj, jelikož podporuje offline funkcionalitu. Aplikace je však také dostupná skrz webové rozhraní, a proto je možné ji využívat i na počítači. Hotová aplikace je v mobilním telefonu dostupná pomocí Neptune Mobile klienta, případně je možné vygenerovat nativní aplikaci. Tato platforma má možnost stáhnout již předpřipravené aplikace z katalogu. Pro použití této platformy na produkčním systému je nutné vlastnit licenci, která je vázána na počet uživatelů.

SAP Fiori je oficiální řešení od společnosti SAP, které nabízí uživatelské rozhraní ve webovém prohlížeči. Díky využití SAPUI5 frameworku se snaží sjednotit vzhled aplikací a využít přizpůsobivého rozhraní, které je možné použít i v mobilních telefonech za pomoci SAP Fiori

aplikace, případně vytvoření nativní aplikace. Fiori také nabízí předpřipravené aplikace, které je možné stáhnout z SAP Fiori knihovny. Fiori oproti Neptunu je řešení, které je možné získat zdarma. Každá aplikace je však součástí balíčku, který je nutné nainstalovat na server. Většina aplikací, které jsou součástí SAP Fiori knihovny, jsou určené pro SAP S4Hana a není možné je implementovat pro SAP Business Suite, který se ve společnosti využívá. Tato platforma bohužel nepodporuje nativní offline funkcionalitu, a proto není vhodné toto řešení využívat pro aplikace, kdy nemá uživatel přístup k internetu.



Obrázek 2: SAP Fiori On-Premis vs SAP Fiori Cloud [3]



## 4 Řešení zadaných úkolů

### 4.1 Aplikace nebo funkční moduly pro SAP

#### 4.1.1 Upravení funkce pro zobrazování faktur z archívu

Tato problematika je svázána se zavedením archivace do systému SAP. Každý dokument má určitou retenční dobu, po kterou musí být dostupný, zpravidla se jedná o několik let. Pro každý dokument se však jedná o jinou retenční dobu podle platné legislativy daného státu a typu dokumentu. Dokumenty, ke kterým se nepřístupuje denně, je možné archivovat, tj. přesunout je do archivačního systému. Výhodou je, že tyto dokumenty jsou stále dostupné, je možno k nim dále přistupovat a zároveň nezahlcují původní systém.

Z důvodu nekompatibility archivačního systému Opentext s platformou Windows server 2003, na které byly nainstalován SAP GUI klient, jsem dostal za úkol vyvinout funkci, která by byla rozšířením standardní funkce s cílem zajistit správné zobrazování dokumentu i na platformě Win2003. Systémové řešení by bylo provést upgrade klientské platformy na poslední verzi Windows, na to ovšem nebyl čas, neboť nekompatibilita se projevila pouze dva týdny před nasazením archivního řešení. Cílem bylo navrhnout řešení s ohledem na souvislost s nestandardními tabulkami pro stažení datového streamu a následnému zobrazení. Od mého konzultanta jsem dostal několik tipů na tabulky s daty, které jsem měl analyzovat a najít mezi nimi relace.

Standardní funkce využívala tabulky TOA01 až TOA03 a TOAHR, která obsahovala číslo a typ dokumentu, ale také repositář a koncovku. Tyto standardní tabulky využívaly standardní funkce správně, bylo však nutné doprogramovat část s využitím tabulek ZZPOD\_TOA01, ZZFI\_COMPANY a ZWFIV\_IIL, které však mají stejnou strukturu. Zde narážím na další nesystémovou část zákaznického řešení, při jehož vývoji nebyly před lety dodrženy zásady normalizace databáze a zákaznické řešení ukládalo duplicitní data do různých tabulek, což pak vedlo k datovým nekonzistencím. Bylo potřeba rozpoznat, zdali je daný dokument zobrazen pomocí výchozího prohlížeče v systému SAP, nebo zvolit mnou navrženou alternativu. Proto bylo vhodné nejdříve projít tabulku ZZPOD\_TOA01 a určit, které řešení využít. Mé řešení pro otevírání nestandardních dokumentu spočívalo ve stažení původních dat do složky TEMP s vygenerovaným názvem podle času na klientském počítači a přidání správné koncovky. Toto řešení bohužel nebylo úplně ideální, jelikož bezpečnostní politika v SAP několikrát zahlásí, že dojde ke stažení a otevření souboru. Všechna tato upozornění musí klient odsouhlasit, řešení je však funkční. Naštěstí se tento problém dá obejít pouze vypnutím upozornění v příslušném klientovi.

#### 4.1.2 Aplikace pro migraci původního repositáře do nového repositáře

Cílem bylo vytvořit aplikace, která dokázala předem známé dokumenty přesunout z jednoho archívu do druhého.



Obrázek 3: Migrace repositáře - vlastní zpracování

Tato funkce byla použita jak při migraci dokumentů z důvodu přechodu na jinou platformu, tak pro migraci dat prodaných částí společnosti. Zároveň bylo nutné všechny tyto akce logovat pro případný audit. Zadání ze strany zákazníka byla pouze výběrová obrazovka, kde uživatel pracující s aplikací, musí určit kategorii, repositář a čísla dokumentů, které chce přesunout. Dále jsem do aplikace přidal 2 možnosti. První možnost sloužila pro zkontrolování obou repositářů, aby nedošlo k přepsání dokumentu již existujícího v cílovém archívu. Druhá varianta dávala možnost změnit tabulku s odkazy na archív v SAPu v případě, že došlo k úspěšnému překopírování.

The screenshot shows the 'ZBC\_MIGRATE\_REP' application window. It has a title bar with the text 'ZBC\_MIGRATE\_REP' and a green checkmark icon. Below the title bar, there is a text label 'For selection P\_DOCS use PHIO\_ID from table DMS\_PH\_CD1'. The main form contains several input fields and checkboxes. The 'PHIO\_IDs' field is highlighted with a yellow background and has a red border. To its right is a 'to' label and another input field. Below these are four input fields: 'Repository destination', 'Repository source', 'Storage category destination', and 'Storage category source'. Each of the first two has a checked checkbox. At the bottom, there are two more checkboxes: 'Diff rep. src and dst' and 'Change table DMS\_PH\_CD1', both of which are also checked.

Obrázek 4: Ukázka úvodní obrazovky programu pro migraci repositářů - vlastní zpracování

Při každé práci s repositáři bylo nutné zachovat i původní záznamy pro případné vrácení změn. Jednou z možností, jak udělat zálohy, bylo změnit těmto dokumentům atribut MANDT. Jedná se o atribut, který nám říká, pro jakého klienta je záznam určen. Stačilo tedy všem úspěšně přeneseným záznamům změnit atribut z produkční hodnoty (070) na námi nově určenou (999). Tím již nebyl záznam dostupný v produkčním klientu pro uživatele, kteří v systému pracují, ale bylo možné jej dohledat pomocí dotazu do databáze a případně záznam obnovit. Pro všechny operace jsem tedy používal příkaz UPDATE místo DELETE, aby nedošlo ke ztrátě dat. Všechny

tyto akce bylo nutné pro případný audit zapisovat do aplikačního logu, aby bylo možné lehce dohledat, který dokument byl přesunut. Aplikační log je poté dostupný pod předem určeným objektem pomocí transakčního kódu SLG1.

Date/Tim...	Number	External ID	Object text	Sub-object text	Transac...	Program	Mode
16.02..	30.132	ZBC_MIGRATE...	Testing applica...		SE38	ZBC_MIGR...	Dialog pro...
16.02..	0		PERSONAS	DYNP	SE38	ZBC_MIGR...	Dialog pro...
16.02..	30.132	ZBC_MIGRATE...	Testing applica...		SE38	ZBC_MIGR...	Dialog pro...
16.02..	0		PERSONAS	DYNP	SE38	ZBC_MIGR...	Dialog pro...
16.02..	30.132	ZBC_MIGRATE...	Testing applica...		SE38	ZBC_MIGR...	Dialog pro...
16.02..	0		PERSONAS	DYNP	SE38	ZBC_MIGR...	Dialog pro...
16.02..	30.132	ZBC_MIGRATE...	Testing applica...		SE38	ZBC_MIGR...	Dialog pro...
16.02..	0		PERSONAS	DYNP	SE38	ZBC_MIGR...	Dialog pro...
16.02..	2.824	ZBC_MIGRATE...	Testing applica...		SE38	ZBC_MIGR...	Dialog pro...
16.02..	0		PERSONAS	DYNP	SE38	ZBC_MIGR...	Dialog pro...
16.02..	2	ZBC_MIGRATE...	Testing applica...		SE38	ZBC_MIGR...	Dialog pro...
16.02..	0		PERSONAS	DYNP	SE38	ZBC_MIGR...	Dialog pro...
16.02..	1	ZBC_MIGRATE...	Testing applica...		SE38	ZBC_MIGR...	Dialog pro...
16.02..	0		PERSONAS	DYNP	SE38	ZBC_MIGR...	Dialog pro...
16.02..	495	ZBC_MIGRATE...	Testing applica...		SE38	ZBC_MIGR...	Dialog pro...
16.02..	0		PERSONAS	DYNP	SE38	ZBC_MIGR...	Dialog pro...

Ty...	Message Text	LTxt
	Document 00505614045f1ed5b4b1ac41ab63816f was successfully copied, line 1	
	Document 00505614045f1ed5b4b1ac41d724616f was not successfully copied, line 2, duo error code 4	?
	Document 00505614045f1ed5b4b1ac41d727616f was not successfully copied, line 3, duo error code 13	?
	Document 00505614045f1ed5b4b1ac42260be16f was successfully copied, line 4	
	Document 00505614045f1ed5b4b1ac42260cc16f was successfully copied, line 5	

Obrázek 5: Zobrazení logu v transakci SLG1 - vlastní zpracování

#### 4.1.3 Aplikace pro úpravu špatných odkazů na dokumenty

Tento program bylo nutné vytvořit z důvodu nekonzistence tabulky TOA01 způsobené předchozím nekorektním návrhem databáze zákazníkem. Systémové řešení by bylo normalizace databáze, které by však bylo daleko časově, ale i zdrojově náročnější, proto bylo nutné vytvořit aplikaci, která otestuje přílohu v dokumentu a opraví odkazy. Některé z dokumentů měly více než jednu přílohu, která je stejného typu. Většinou je však jeden z těchto souboru nevalidní a nedostupný. Jelikož je tabulka opravdu velká, proto jsem jako vstup vytvořil textové pole pro dokumenty (select-option). Později bylo nutné přidat možnost filtrování tabulky nejen podle čísla dokumentu, ale také podle typu objektu. Pro každý takovýto dokument jsem našel počet existujících referencí. Podle počtu referencí jsem měl několik možností, jak dále pokračovat. V případě, že program najde více než 2 záznamy stejného typu, je nutné zkontrolovat, který z těchto záznamů je nedostupný. K tomuto účelu jsem našel funkci, která je schopna se připojit na archiv a zjistit, jestli je soubor dostupný, případně vrátit jeho typ a velikost.

Test for function group	ARCHIVOBJECT
Function module	ARCHIVOBJECT_STATUS
Uppercase/Lowercase	<input type="checkbox"/>
Runtime:	511.873 Microseconds
Import parameters	Value
ARCHIV_DOC_ID	00505614045F1EE6A2DEB764E10BC374
ARCHIV_ID	SI
SIGN	
DOC_TYPE	
SIGNATURE	X
Export parameters	Value
ARCHIV_OBJECT_STATUS	ONLINE
AR_DATE	20161005
AR_TIME	121749
DOCUMENT_TYPE	PDF

Obrázek 6: Ukázka volání funkce pro zjištění stavu dokumentu - vlastní zpracování

Stačilo zjistit, jestli dokumenty, které jsem načetl, jsou dostupné, případně který z nich je nedostupný. Původní záznam jsem přesunul pod neexistujícího klienta 999 pro zachování původního záznamu a případnou obnovu. Zjistil jsem, že úpravy v původní tabulce je nutné provést i v nestandardní tabulce, aby nedošlo k nekonzistenci dat. V nestandardní tabulce muselo dojít ke změně informací o objektu a záznam nahrát. Všechny tyto akce jsem musel zapisovat do aplikačního logu. Poprvé jsem zde využil předpřipravených textových zpráv. Abych nemusel stále vypisovat danou zprávu, je možné v transakčním kódu SE91 vytvořit předpřipravené zprávy, na které se dá odkazovat. Funkce už automaticky na předem určená místa doplní text, který si zvolím.

Message class

ZBC

Activ.

Attributes

Messages

</

Obrázek 7: Transakční kód SE91 pro správu a předpřipravených zpráv - vlastní zpracování

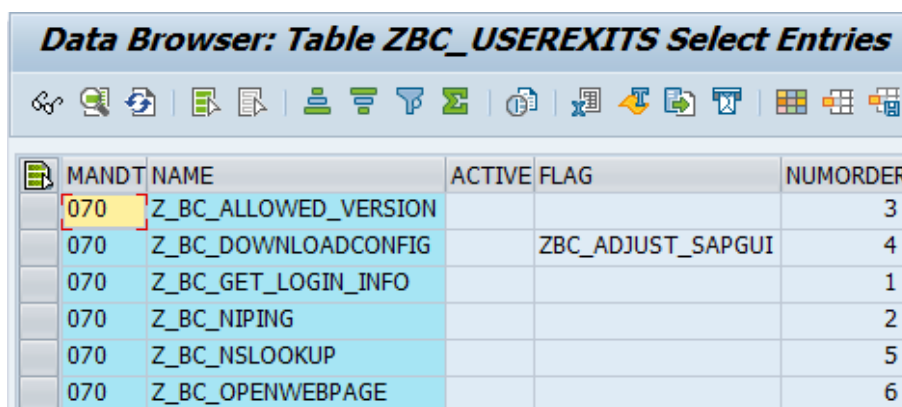
#### 4.1.4 Automatické spuštění funkcí po přihlášení a jejich efektivní správa

Při přihlášení uživatele bylo nutné zajistit spuštění různých funkcí, které např. testovaly připojení od uživatele na server, případně stahovaly konfigurační soubory, nebo hlásily informace o používání staré verze SAP GUI.

K spuštění programu po přihlášení uživatele slouží user exit, který se chová podobně jako program. Tento program však postupně narůstá a není jednoduché v něm snadno vypnout určitou funkci.

Mým úkolem bylo vytvořit řešení pro efektivní povolování částí kódů nebo funkcí, které se mají spustit po přihlášení uživatele. Dříve bylo nutné přidat část kódu přímo do hlavního programu, což mělo několik rizik. Jedno z nich byla vyskytující se chyba v kódu, která mohla způsobit nefunkčnost prostředí, a tím zamezila přihlášení uživatele. Mezi další problém patří to, že funkci nebylo možné jednoduše vypnout či zapnout v libovolnou dobu a v libovolném systému.

Každý program je nutné vyvíjet na vývojářském systému DEV, který se později přesouvá do testovacího systému TST, kde je následně důkladně otestován. Po otestování a schválení je možné, aby se takovýto program přesunul do produkčního systému ENS. Tento proces ve korporaci Stora Enso se děje jednou týdně. Moje řešení spočívalo ve vytvoření jednoduché tabulky, která obsahovala název funkce, pořadí a příznak, aby se daná funkce spustila. Později jsem přidal ještě jeden příznak, aby bylo možné určit, kterým uživatelům se má funkce spustit. Tabulku jsem vytvořil jako CLIENT SPECIFIC dle doporučení SAP. V případě problému s jakýmkoliv modulem je možné se přihlásit do referenčního klienta, kde se tyto funkce nespouštějí a problémový modul lze smazat. Tato oprava je daleko efektivnější než složité posílání oprav celého kódu v programu.



MANDT	NAME	ACTIVE	FLAG	NUMORDER
070	Z_BC_ALLOWED_VERSION			3
070	Z_BC_DOWNLOADCONFIG		ZBC_ADJUST_SAPGUI	4
070	Z_BC_GET_LOGIN_INFO			1
070	Z_BC_NIPING			2
070	Z_BC_NSLOOKUP			5
070	Z_BC_OPENWEBPAGE			6

Obrázek 8: Tabulka pro správu funkcí po přihlášení uživatele v transakčním kódu SE16 - vlastní zpracování

Vytvořil jsem tedy hlavní program, který se spustí každému uživateli po přihlášení. Tento program načte všechny aktivní funkční moduly z tabulky a začne je postupně spouštět. Nejdříve ověří, zdali opravdu takovýto modul existuje v systému, aby nedošlo k pádu aplikace. V případě,

že funkce existuje, dojde ke spuštění. Součástí tohoto programu bylo i vytvoření několika funkcí, které nejsou předmětem této práce.

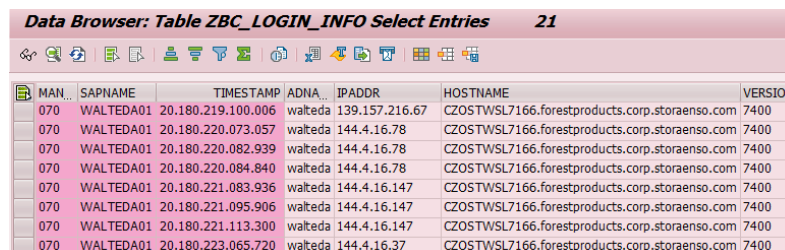
#### 4.1.5 Aplikace pro audit a monitoring uživatelů

Samotný SAP systém disponuje několika možnostmi pro monitoring. Jedna z možností je využít např. transakční kód SM20. Bohužel standardní prostředky pro monitoring umožní filtrování pouze na vybraného uživatele, proto bylo nutné vytvořit monitoring podle našich potřeb. Tento monitoring bylo nutné zavést nejen kvůli migrace datového centra k novému poskytovateli infra služeb, ale také kvůli defraudaci v Číně, kdy docházelo ke sdílení jmenných účtů.

Z důvodu migrace datového centra k novému poskytovateli infra služeb, bylo nutné zjistit, kolik uživatelů se připojuje do sítě mimo korporátní síť za pomoci např. VPN tunelu. Všechny tyto uživatele bylo nutné upozornit na změny IP adres a přizpůsobit tomu i firewall. Tito uživatelé navíc nepoužívají globální konfigurační soubor, a bylo tedy nutné je upozornit na změny, které musí ručně v SAP GUI nastavit. Zaměstnanci využívají globální konfigurační soubor, který je dostupný odkudkoliv z korporátní sítě, případně za použití VPN, která je jim poskytnuta.

Základní analýza zahrnovala uživatelské jméno do systému SAP a Active Directory, datum a čas přihlášení a uživatelskou IP a hostname. Později jsem přidal možnost pro zjištění verze SAP GUI, abychom měli přehled, kolik uživatelů používá konkrétní verzi, a tím jsme získali oblasti, ve kterých je nutné udělat upgrade SAP GUI.

Program byl rozdělen na 2 části. V první části bylo nutné vytvořit funkci, která se stará o sběr informací od každého uživatele. Modul jsem navrhl tak, aby byl spuštěn vždy, když se uživatel přihlásí do systému, a tyto informace jsem poté ukládal do předem vytvořené tabulky. Datum a čas jsem ukládal ve tvaru timestamp, jednak abych ušetřil místo v tabulce, jednak aby se dalo v tabulce lépe filtrovat. Druhá část zahrnovala vytvoření programu, který nám zpřístupní výsledky dat v přehledném formátu. Vytvořil jsem program, který všechny uložené informace zobrazí v přehledné tabulce, které se říká ALV grid. Tato tabulka dovolí běžnému uživateli filtrovat a třídit data bez jakýchkoliv technických dovedností. Dále jsem z důvodu přehlednosti ke každému uživateli přiřadil informaci s jeho jménem, příjmením a personálním číslem, také jsem převedl timestamp na políčko pro datum a čas. Ve výsledném programu se dalo velmi snadno filtrovat a lehce určit, který uživatel se v určitý den přihlásil a kdy došlo k aktualizaci klienta.

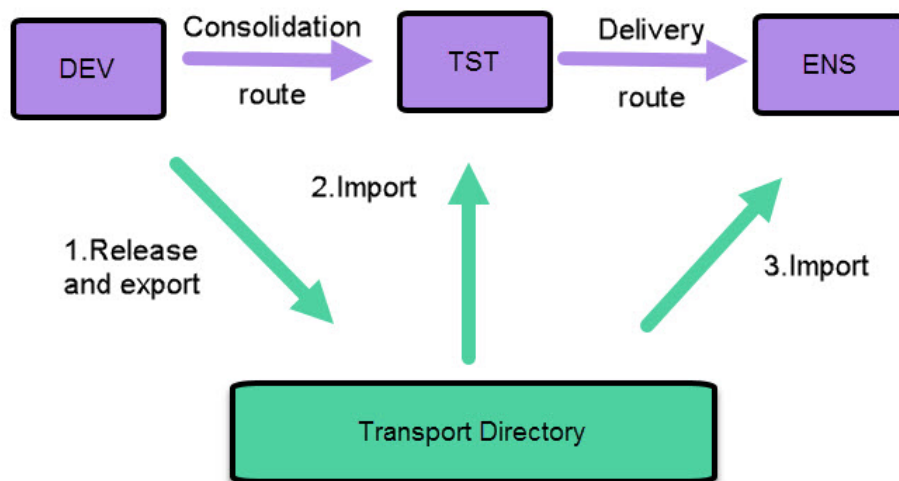


MAN.	SAPNAME	TIMESTAMP	ADNA.	IPADDR	HOSTNAME	VERSION
070	WALTEDA01	20.180.219.100.006	walxeda	139.157.216.67	CZOSTWSL7166.forestproducts.corp.storaenso.com	7400
070	WALTEDA01	20.180.220.073.057	walxeda	144.4.16.78	CZOSTWSL7166.forestproducts.corp.storaenso.com	7400
070	WALTEDA01	20.180.220.082.939	walxeda	144.4.16.78	CZOSTWSL7166.forestproducts.corp.storaenso.com	7400
070	WALTEDA01	20.180.220.084.840	walxeda	144.4.16.78	CZOSTWSL7166.forestproducts.corp.storaenso.com	7400
070	WALTEDA01	20.180.221.083.936	walxeda	144.4.16.147	CZOSTWSL7166.forestproducts.corp.storaenso.com	7400
070	WALTEDA01	20.180.221.095.906	walxeda	144.4.16.147	CZOSTWSL7166.forestproducts.corp.storaenso.com	7400
070	WALTEDA01	20.180.221.113.300	walxeda	144.4.16.147	CZOSTWSL7166.forestproducts.corp.storaenso.com	7400
070	WALTEDA01	20.180.223.065.720	walxeda	144.4.16.37	CZOSTWSL7166.forestproducts.corp.storaenso.com	7400

Obrázek 9: Ukázka uložených informací o uživateli WALTEDA01 - vlastní zpracování

#### 4.1.6 Aplikace pro kontrolu transportu

Ve společnosti jsem pracoval hlavně ve 3 systémech. Jedná se o systémy Development(DEV), Testing(TST) a Production(ENS). Vývoj je možné dělat pouze v systému DEV, ostatní systémy jsou uzamčeny pro úpravy. Standardní postup je transportovat hotový program do systému TST, kde je testován a zdokumentováno, zdali všechny požadované funkce pracují správně. Po schválení je možné, aby byl transport odeslán do produkčního systému. Tento proces se provádí jednou týdně, ale může se lišit podle společnosti. Při vývoji se častokrát stávalo, že probíhal paralelní vývoj několika projektů, které o sobě nevědí, ale provádějí vývoj a úpravy ve stejné funkci. Po nahrání transportu s funkcí v obráceném pořadí dojde k přepsání na starší verzi, případně downgradu programu. Tento problém často způsobil úplný pád aplikace, který byl poté těžko dohledatelný, proto jsme byli nuceni najít řešení, které by dopředu dokázalo detekovat riziko a odhalit tento problém. V případě, že se naskytne problém a detekuje se riziko, je nutné kontaktovat daného uživatele, případně vývojáře, zda si je vědom možného rizika a zda s ním souhlasí.



Obrázek 10: Ukázka transportu - vlastní zpracování

Základem bylo vymyslet princip, jak kontrolovat transporty, které jsou v produkčním systému. K tomuto účelu jsem využil možnost RFC, která umožňuje spustit funkci, nacházející se v jiném systému. Jeden transport může obsahovat mnoho objektů. Jelikož všechny změny musí být v systému DEV a až poté mohou být v produkčním systému, stačí zjistit, v jakém transportu je daný objekt, a získávat unikátní čísla transportu v tabulce, která poté mohou zpracovávat v DEV systému. Tím se sníží náročnost operací v produkci na minimum, jelikož všechny operace zpracovávám jen v jednom systému. Jako vstup programu jsou čísla všech transportů, která se budou transportovat do produkčního systému. Je potřeba zadat přesnou sekvenci

transportů tak, jak se budou importovat. Jeden z důvodu je, aby v systému existovaly všechny komponenty, které jsou potřebné pro aktivaci a spuštění aplikace.

V systému SAP máme několik základních typu transportu. První z nich se nazývá Workbench request, který slouží pro transportování objektu jako jsou programy, funkční moduly a další objekty, které jsou součástí programu. Druhý se nazývá Customizing request. Tento transport obsahuje hlavně tabulky se změnami, které se transportují do jiného systému. Oba typy jsou po vytvoření ve stavu Modifiable a pro přesun do dalšího systému je nutné ho převést do stavu Released, kdy dojde k zamčení objektů a přípravě pro export.

V mém programu bylo nejdříve potřeba zjistit, zda-li všechny zadané transporty jsou ve stavu „Release“ a je možné je transportovat a zkontrolovat.

Jako vstup pro program byl seznam, který obsahuje čísla transportu, systém, vůči kterému se budou objekty testovat, a seznam objektů, které se mají ignorovat. Většinou je potřeba ignorovat nově vygenerované role, případně jiné tabulky s autorizacemi. Prvním krokem bylo načíst všechny objekty ze zadaných transportů a uložit si je do interní tabulky. V případě, že objekt již v tabulce je, stačilo pouze změnit název transportu a zaznamenat vždy jen poslední změnu. V tomto kroku jsme již mohli vzdáleně použít funkci, která vrací seznam dostupných transportů v produkčním systému. Poté je možné jednotlivé transporty procházet jeden po druhém a kontrolovat datum a čas vytvoření. Je-li transport, který se chystá být importován do produkčního systému, starší než v danou chvíli dostupný transport, je nezbytné upozornit na událost. Konečným výstupem je tabulka s informacemi o potencionálním riziku, uživateli, který transport vytvořil, včetně jeho e-mailu. Uvedení e-mailové adresy u výstupu nám pomáhá k další komunikaci s uživatelem.

Během vývoje se však naskytlo několik dalších problémů, které bylo nutné opravit. Jedním z problémů bylo, že některé tabulky obsahovaly více klíčů, a program tak vyhodnotil, že se jedná o chybu, přičemž šlo o falešný poplach. Při mých úvahách nad tímto problémem jsem zjistil, že je nutné procházet ještě další tabulku, a to v případě, že se jedná o objekt typu customizing, který je součástí kontrolovaného transportu.

V případě, že srovnávané customizing objekty a jejich klíče se neshodovaly, nehrozilo žádné další riziko. Jako výstup celého programu dle zadání jsem využil ALV grid, se kterým jsem měl už předchozí zkušenosti.

Status	TR for release	Description of trans	TR Deployed	Description of trans	Object	Owner	Owner email
OK	DEVK993753	IP_CR2_9_MM_Enhancement - Impr.			DEVK993754 20180122 122525 AC...	WALTEDA01	daniel.walter@storaenso.se
FAILED	DEVK993753	IP_CR2_9_MM_Enhancement - Impr.	DEVK993850	IP_CR2_9_MM_Enhancement - Impr.	ZSES0_DOC_SHIPMENT	WALTEDA01	daniel.walter@storaenso.se
OK	DEVK993753	IP_CR2_9_MM_Enhancement - Impr.			LZSES0_MULTILEGFO1	WALTEDA01	daniel.walter@storaenso.se
FAILED	DEVK993753	IP_CR2_9_MM_Enhancement - Impr.	DEVK994122	IP_CR2_9_MM_Enhancement - Impr.	ZCL_SESD_MULTILEG_IMPROVEMENT	WALTEDA01	daniel.walter@storaenso.se
OK	DEVK993753	IP_CR2_9_MM_Enhancement - Impr.			ZSES0_D_SHIPP_DOC_TYPE	WALTEDA01	daniel.walter@storaenso.se
OK	DEVK993753	IP_CR2_9_MM_Enhancement - Impr.			ZSES0_D_SHIPP_DURATION	WALTEDA01	daniel.walter@storaenso.se
OK	DEVK993753	IP_CR2_9_MM_Enhancement - Impr.			ZSES0_E_DESTINATION_PLANT	WALTEDA01	daniel.walter@storaenso.se
OK	DEVK993753	IP_CR2_9_MM_Enhancement - Impr.			ZSES0_E_SHIPP_DOC	WALTEDA01	daniel.walter@storaenso.se
OK	DEVK993753	IP_CR2_9_MM_Enhancement - Impr.			ZSES0_E_SHIPP_DOC_TYPE	WALTEDA01	daniel.walter@storaenso.se
OK	DEVK993753	IP_CR2_9_MM_Enhancement - Impr.			ZSES0_E_SHIPP_DURATION	WALTEDA01	daniel.walter@storaenso.se

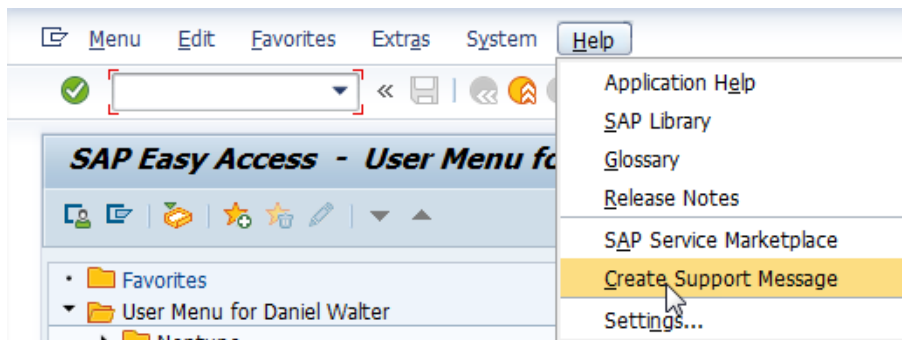
Obrázek 11: Ukázka výstupu programu - vlastní zpracování



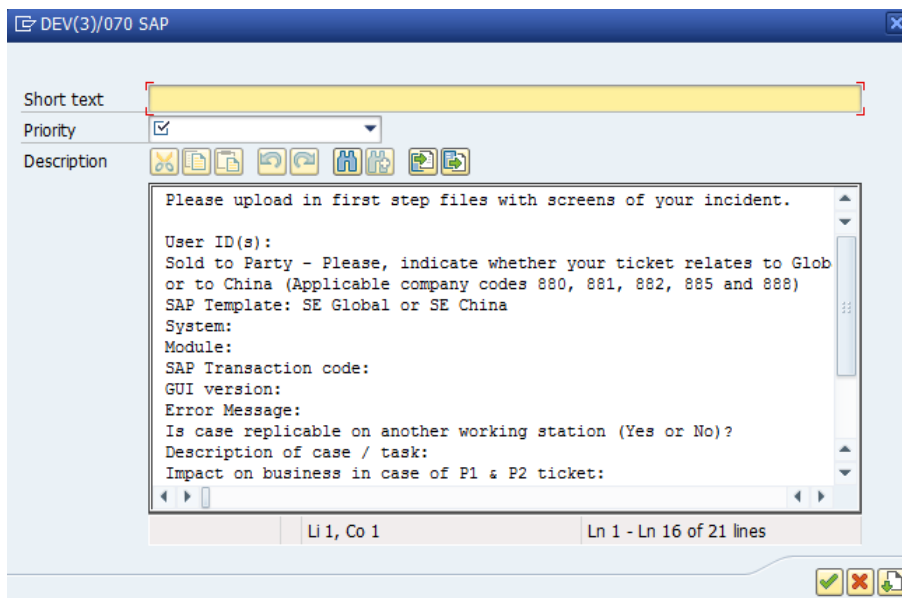
#### 4.1.7 Aplikace pro zasílání ticketu ze systému SAP

Nativní řešení pro tickety v systému SAP je systém Solution Manager. Z důvodu přechodu na řešení Service Now bylo nutné předělat posílání všech ticketů ze systému SAP do tohoto systému.

U této aplikace bylo potřeba přidat možnost vytváření ticketu nebo dát uživateli možnost nahlásit problém přímo z aplikace SAP. Aplikace má za cíl po požadované akci vyzvat uživatele, aby popsal problém, který mu vznikl. Při odeslání se v systému ServiceNow vytvoří událost i s potřebnými informacemi, která nám dále pomůže vyřešit daný incident. Dřívější řešení fungovalo tak, že uživatel musel ticket vytvářet ručně v systému Service Now. Řešení, na kterém jsem pracoval, umožňuje nahlásit incident přímo ze systému SAP. Při nahlášení incidentu se odešle email ze systému SAP, a to v přesně definovaném formátu, který systém ServiceNow sám identifikoval, roztrídil a upozornil tým, který byl určen k posouzení daného incidentu.



Obrázek 12: Vytvoření nového ticketu do systému Service Now - vlastní zpracování



Obrázek 13: Dialog pro zasílání ticketu do nového systému ServiceNow - vlastní zpracování

Pomocí nové aplikace si uživatel sám určí prioritu, stručně popíše problém a doplní do předpřipravené šablony, o jaký problém se jedná, a má zde možnost dopsat další informace. Tyto informace se odešlou do systému Service Now. Uživatel má často problém technicky popsat situaci, která mu nastala, a proto jsou nám bez vědomí uživatele zasílány technické parametry, které nám ulehčují práci při řešení jeho problému. Součástí technických parametrů je také e-mail uživatele, protože systém Service Now přijímá e-maily jen z určité adresy, podle které určí, kterému týmu má ticket přiřadit.

#### 4.1.8 Aplikace pro zasílání idocu

Tato aplikace byla primárně určena pro divizi Bio materiál. Rozhodli jsme, že by bylo dobré, aby aplikace fungovala univerzálně. Hlavním cílem bylo udělat aplikaci, která poběží na pozadí a pro předem určené podmínky bude zapisovat potřebné informace a výsledek zasílat na email nebo zobrazovat přímo na výstupu programu. Záměrem bylo, aby každá divize dostávala informaci o tom, které idocy není možné zpracovat, které skončily s chybou či v určitém status kódu.

Idoc je dokument, který se využívá pro export a import obchodních a logistických dokumentů. Jedná se o nativní formát SAPu, který je však možné konvertovat do jiných formátů.

Z těchto informací je poté možné získat statistiky, případně získat ty idocy, které skončily s chybou, a je nutné je ručně zpracovat, opravit či udělat jiná opatření. Vstupní informace do programu jsem rozdělil do 3 sekcí, aby bylo vše přehledné. V první části je možné zadat základní informace, jako je stav dokumentu a jeho typ, případně omezit určité období. Dále je zde nastavení pro emailovou adresu a informace, jestli má být výstup odeslán na email, či přímo zobrazen v programu. V druhé části je možné zadat až 10 sloupců s informacemi, které se mají zobrazovat..

Tato část souvisí s poslední sekcí, kde je nutné zadat segment a název pole, ze kterého se mají načíst data. Jelikož idocy chodí do systému hromadně, je v této části možné zadat podrobnější podmínku přímo na políčku, a tím získat pouze dokumenty, které jsou určené pro konkrétní divizi. Každá divize, která chce využívat tento program, si může vytvořit svou vlastní variantu s přednastavenými daty pro výstup, který jim bude vyhovovat.

V tomto programu jsem poprvé využil práci s field-symboly. Tato funkce má využití v případě, že potřebujeme pracovat s dynamickými daty. Celý základ byl postaven na dynamické tabulce, kde si uživatel určí formát konečného výstupu. Program prochází všechny idocy, které našel. Každý záznam opětovně zkontroluje a ověří, zda daný uživatel má právo tyto dokumenty zobrazit. K tomuto účelu slouží klíčové slovo Authority check.

---

```
AUTHORITY-CHECK OBJECT 'S_IDOCMONI'
ID 'ACTVT' FIELD '03'
ID 'EDI_DIR' FIELD WA_EDIDC-DIRECT
ID 'EDI_MES' FIELD WA_EDIDC-MESTYP
```

```

IF SY-SUBRC <> 0.
    CONTINUE.
MESSAGE 'Some idoc was skipped, missing authorization.' TYPE 'I'.
ENDIF.

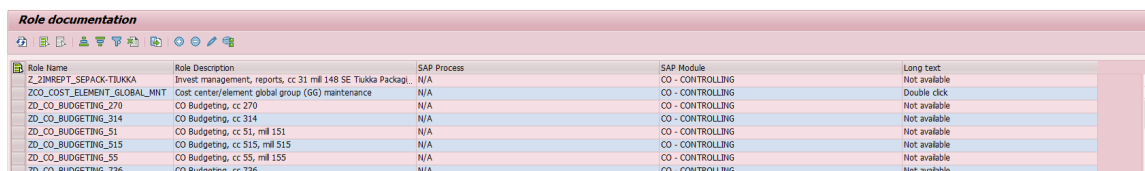
```

#### Výpis 1: Ukázka volání Authority checku

V případě, že uživatel nemá dostatečná oprávnění, je o této skutečnosti informován pomocí informační tabulky a následně dojde k přerušení zpracovávání daného dokumentu. Výsledek je zobrazen uživateli v tabulce, kterou si předdefinuje před spuštěním, nebo mu je výsledek zaslán na email jako příloha formou excelovské tabulky.

### 4.1.9 Aplikace pro dokumentaci rolí

Jednalo se o projekt, který měl za cíl sjednotit technické a business role. Jako business roli můžeme uvést např. pozici člověka, který je ve funkci kontrolora. Díky této aplikaci si každý uživatel může snadno zjistit, o jaké technické role by si měl požádat.



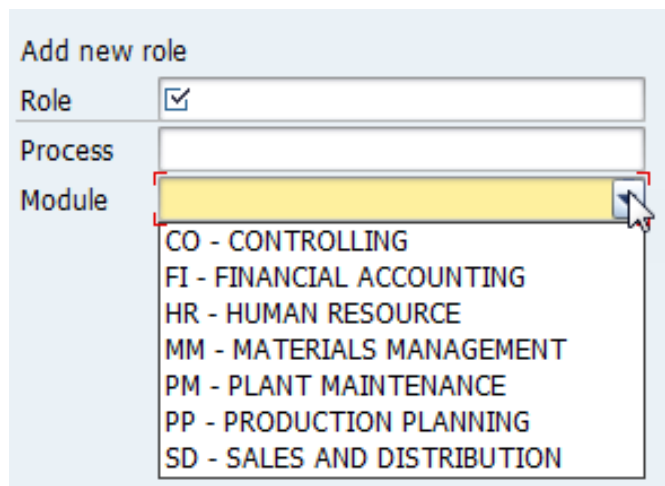
Role Name	Role Description	SAP Process	SAP Module	Long text
Z_3MREPT_SEPACK-TLUKA	Invest management, reports, cc 31, ml 148 SE TLUKA Packag...	N/A	CO - CONTROLLING	Not available
ZCO_COST_ELEMENT_GLOBAL_MNT	Cost center/element global group (GG) maintenance	N/A	CO - CONTROLLING	Double click
ZD_CO_BUDGETING_270	CO Budgeting, cc 270	N/A	CO - CONTROLLING	Not available
ZD_CO_BUDGETING_314	CO Budgeting, cc 314	N/A	CO - CONTROLLING	Not available
ZD_CO_BUDGETING_51	CO Budgeting, cc 51, ml 151	N/A	CO - CONTROLLING	Not available
ZD_CO_BUDGETING_515	CO Budgeting, cc 515, ml 515	N/A	CO - CONTROLLING	Not available
ZD_CO_BUDGETING_55	CO Budgeting, cc 55, ml 155	N/A	CO - CONTROLLING	Not available
ZD_CO_BUDGETING_736	CO Budgeting, cc 736	N/A	CO - CONTROLLING	Not available

Obrázek 14: Zobrazení dostupných technických rolí pro kontrolora - vlastní zpracování

Při vybrání technické role je možné zobrazit transakční kódy, případně organizační data pro danou roli. Každé business roli je přiřazen proces a modul pro přehledné filtrování, případně long text, který po dvojkliku slouží k zobrazení delšího popisu.

Pro administrátory bylo nutné udělat i možnosti pro přidání, úpravu a odstranění záznamu. Tyto funkce jsou dostupné jen uživatelům, kteří mají daná oprávnění, k tomu slouží další program, kterému jsou předány potřebná data. Na tomto projektu jsem se naučil pracovat i se screeny. Jedná se o GUI obrazovku, do které můžeme umístit prvky, s kterými poté uživatel může pracovat. Každá aplikace má ve výchozím stavu jen hlavní selection-screen, který má vždy číslo 1000.

Bylo potřeba vytvořit ještě další 3 GUI obrazovky. Každý pro ALV tabulku, první pro výchozí tabulku s rolemi, druhou pro transakční kódy a poslední s organizačními daty. Toto řešení jsem zavedl z důvodu velkého množství dat. V případě využití jen jedné obrazovky by bylo nutné data vždy znova načítat, a celá aplikace by byla zbytečně pomalá. I v druhém programu bylo nutné vytvořit další GUI obrazovku pro úpravu nebo přidání nové role. Posledním krokem bylo vytvořit možnost, jak naimportovat data z existujícího excel souboru, který se do té doby používal. K tomuto účelu jsem přidal další dialogové okno pro nahrávání z předem určené šablony, kterou je možno z programu stáhnout.

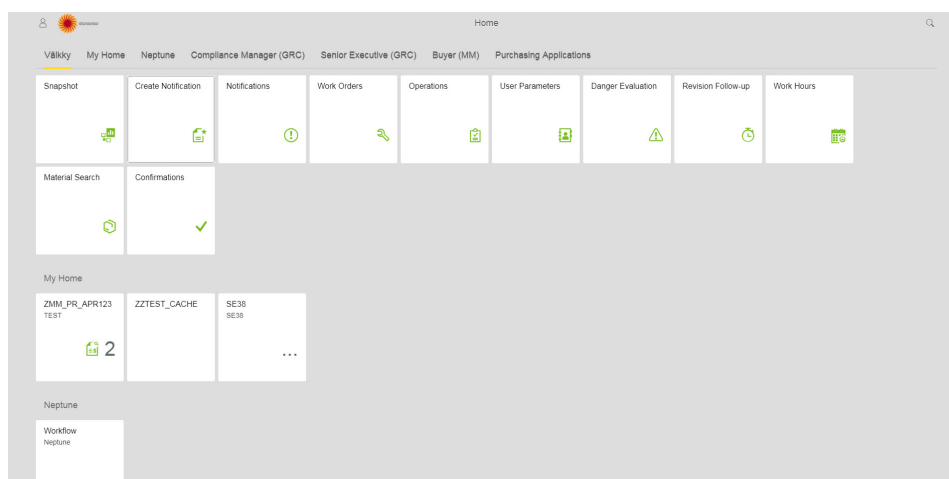


Obrázek 15: Ukázka přidávání nového záznamu s možnostmi modulů - vlastní zpracování

## 4.2 Mobilní vývoj

Následující část budu věnovat mobilnímu vývoji pro informační systém. Mobilní aplikace bylo nutné vytvořit, aby hlavně top management společnosti nemusel zapínat počítač pro činnosti, které mohou vyřídit daleko rychleji za pomoci telefonu, např. z důvodu schvalování cestovních náhrad, případně schvalování nejrůznějších faktur. Jedná se o pohodlné řešení, kdy mohou zaměstnanci pracovat při cestování. Mezi další požadavek patří offline funkcionality, aby bylo možné tyto akce provádět i mimo dosah signálu bez připojení k internetu, např. v letadle. Na výběr bylo možné vytvářet vývoj na platformě Neptune nebo Fiori.

Obě tato řešení mají hlavní rozhraní, které se nazývá launchpad, ze kterého si uživatelé mohou spustit aplikace, ke kterým mají oprávnění. Hlavně potom v SAP Fiori si každý uživatel může nastavit oblíbené položky a přenastavit menu podle svých potřeb.



Obrázek 16: Ukázka SAP Fiori Launchpadu - vlastní zpracování

V případě, že by se Fiori využívalo standardně, bylo by možné přistupovat k tomuto rozhraní ze všech zařízení ke sjednocenému rozhraní, které si uživatel předem vytvoří.

Při svém vývoji jsem hlavně pracovat s platformou Neptune, která je ve společnosti implementována na každém systému se sjednoceným front-end i back-endem. Tomuto řešení se nazývá On-Premise. Platforma SAP Fiori je už implementována zvlášť, a tedy na jednom serveru běží front-end rozhraní, které se poté spojuje s dalším systémem back-endem, odkud stahuje data. SAP Fiori má dále možnost implementace s pomocí SAP Fiori Cloud, kde se využívá veřejný server SAP Fiori pro Front-end server a spojuje se s druhým back-end systémem, který je uvnitř interní sítě. K tomuto řešení je nutné implementovat SAP Cloud Connectoru.

SAP dokáže komunikovat s externími aplikacemi pomocí webových služeb. Většina služeb využívá Open Data Protokol (OData). Jedná se o protokol, který je založený na architektuře REST a standardizován podobně jako XML či JSON. Dává nám možnost se dotazovat na určitá data pomocí předem definovaných parametru v URI. Výstupní data je možno filtrovat a získat data a přesný výstup v požadovaném formátu tak, jak jej vyžadujeme.

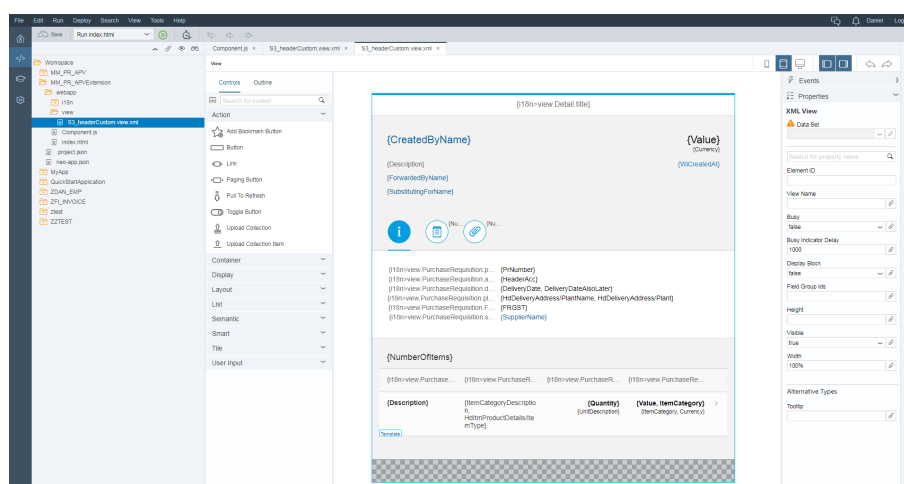
---

```
<feed xmlns="http://www.w3.org/2005/Atom" xml:base="/sap/opu/odata/sap/
  ZE_VLK_BACKEND_SRV/">
  <title type="text">SalesDocSet</title>
  <updated>2018-03-31T20:29:54Z</updated>
  <author>
    <name firstname="Daniel" lastname="Walter">
  </author>
  <entry>
    <id>SalesDocSet('40002356')</id>
    <updated>2018-03-31T20:29:54Z</updated>
    <category term="ZE_VLK_BACKEND_SRV.SalesDoc" />
    <content type="application/xml">
      <properties xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices">
        <d:SalesOrd>40002356</d:SalesOrd>
        <d:SalesOrdDescr>Tainionkoski Yhteiset EKP</d:SalesOrdDescr>
        <d:ValidFrom>2009-01-01T00:00:00</d:ValidFrom>
        <d:ValidTo>2020-12-31T00:00:00</d:ValidTo>
        <d:DocType>ZSCC</d:DocType>
      </properties>
    </content>
  </entry>
</feed>
```

---

Výpis 2: Ukázka OData feedu z webové služby

Tyto služby je možné vytvořit v transakčním kódu SEGW (SAP NetWeaver Gateway Service Builder). V této transakci je možné vytvořit službu a nastavit metody, které můžeme externě volat. Tyto služby je nutné vytvářet hlavně při vytváření aplikace na platformě SAP Fiori, kdy je nutné mapovat akce na události ve webové službě. SAP Fiori funguje primárně jako webové rozhraní a uživatel má možnost i toto rozhraní využít na mobilním telefonu. K tomuto účelu funguje oficiální aplikace, kterou si může kdokoli stáhnout. Pro fungování této aplikace je nutné zadat firemní server a poté se stačí přihlásit. Vývoj pro platformu SAP Fiori je značně složitější než platforma Neptune, se kterou jsem pracoval častěji. Pro vývoj je také nutné využívat Cloud a spojit SAP Cloud s firemním serverem za pomoci SAP Cloud Connectoru.



Obrázek 17: Webové prostředí WebIDE pro vývoj na platformě SAP Fiori - vlastní zpracování

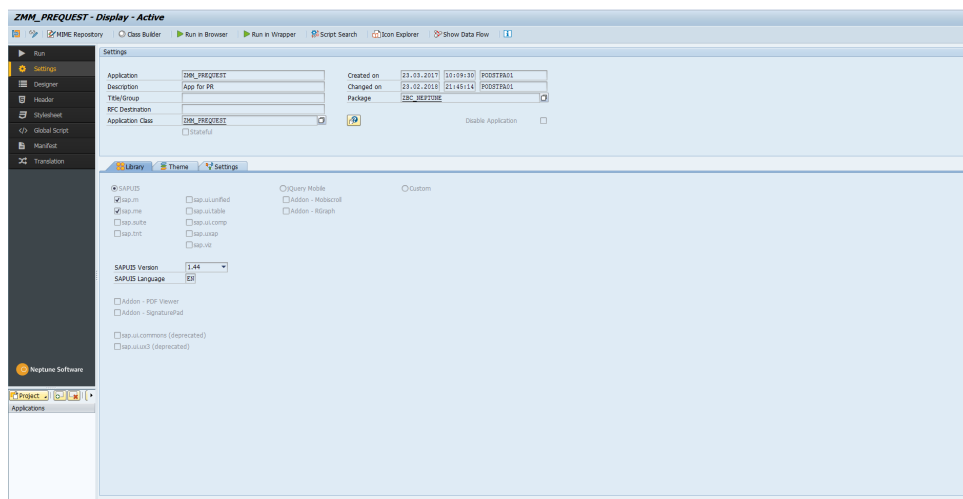
Pro platformu Neptune jsem se rozhodl hlavně z důvodu nativní offline funkcionality, která byla vyžadována pro všechny aplikace. Zároveň je vývoj na tuto platformu z mého pohledu jednodušší a rychlejší.

Obě tato řešení však fungují na stejném frameworku SAPUI5, a proto výsledné aplikace vypadají stejně. Dále z obou aplikací je možné vytvořit nativní aplikaci za pomoci Adobe Phonegap a využívat dostupné knihovny, např. pro načítání barcode kódu. Každé řešení však má své výhody a nevýhody, které jsem srovnal v kapitole 3.2.

### 4.3 Aplikace pro schvalování faktur na platformě SAP Neptune

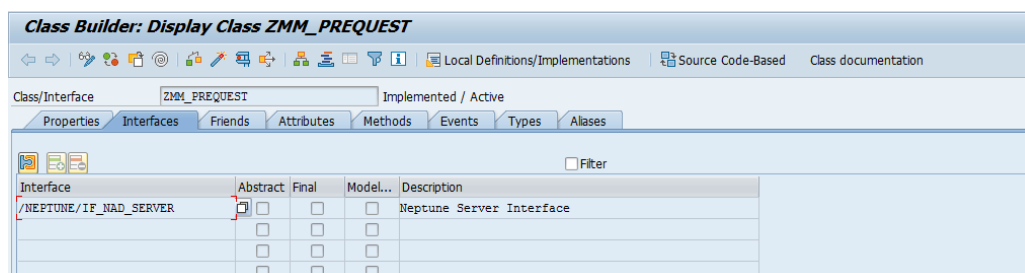
Tato podkapitola bude zaměřena na aplikace pro schvalování faktur, kterou jsem vytvářel pro management. Jeden z požadavků bylo, aby tato aplikace fungovala i bez připojení k internetu. Z tohoto důvodu jsem zvolil platformu Neptune, která má nativní podporu pro offline funkcionality. Tato aplikace ušetří uživatelům mnoho času.

Pro vývoj mobilních aplikací na této platformě funguje předem určená transakce. V této transakci jsem si nejprve vybral knihovny a šablonu, které budu v projektu potřebovat.



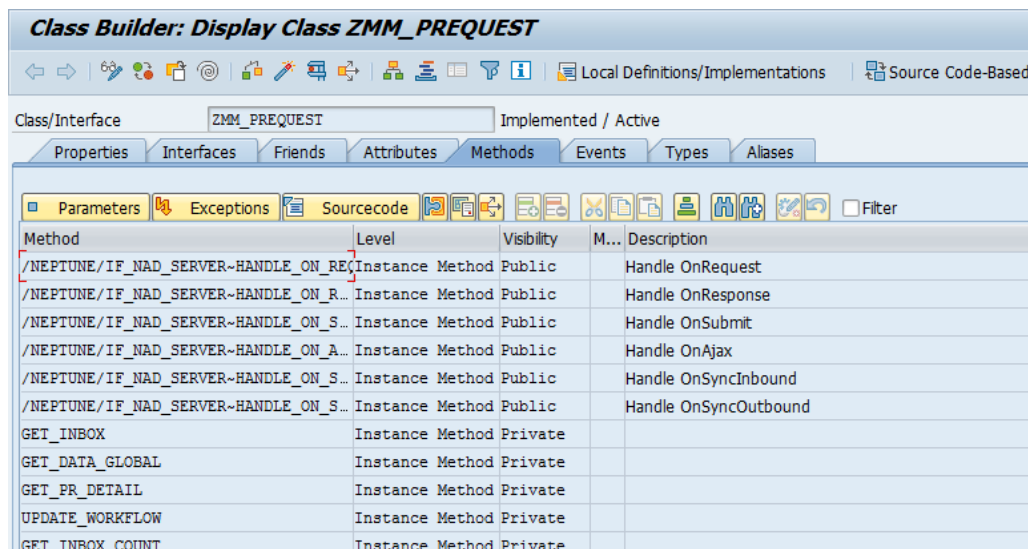
Obrázek 18: Vývojové prostředí na platformě Neptune - vlastní zpracování

Další důležitou funkcí je komunikace grafického rozhraní se službou, která nám poskytne data. Neptune funguje na bázi tříd a každá aplikace je napojená na jednu třídu, se kterou poté komunikuje. Tato třída musí mít naimplementované rozhraní pro tuto platformu, a poté začne třída pracovat podobně jako webová služba. Dále se musí implementovat všechny požadované metody, které se později využijí.



Obrázek 19: Implementace třídy pro komunikaci s platformou - vlastní zpracování

Po implementaci rozhraní se do třídy automaticky přidávají metody, které jsou ihned dostupné, a můžeme je implementovat podle potřeb.



Obrázek 20: Implementace třídy pro komunikaci s platformou - vlastní zpracování

Každá z metod je vyvolána při jiné akci. Nejdůležitější je metoda HANDLE\_ON\_AJAX, která se stará o všechna volání v aplikaci, například při stisku přednastaveného tlačítka. Webová služba slouží jako backend a je zde napsána celková logika aplikace.

```
SELECT * FROM HRUS_D2
      INTO TABLE IT_HRUS_D2
      WHERE REP_NAME EQ SY-UNAME
      AND BEGDA <= SY-DATUM
      AND ENDDA >= SY-DATUM
      AND ACTIVE NE SPACE.
```

```
WA_HRUS_D2-US_NAME = SY-UNAME.
APPEND WA_HRUS_D2 TO IT_HRUS_D2.
```

```
CALL METHOD GET_DATA_GLOBAL( ).
```

```
WA_TASK_FILTER-WI_RH_TASK = 'TS96000094'.
APPEND WA_TASK_FILTER TO IT_TASK_FILTER.
```

```
*-----*
* GET DATA
*-----*

WAIT UP TO 2 SECONDS.
CLEAR IT_WF_LIST.
LOOP AT IT_HRUS_D2 INTO WA_HRUS_D2.
```



```

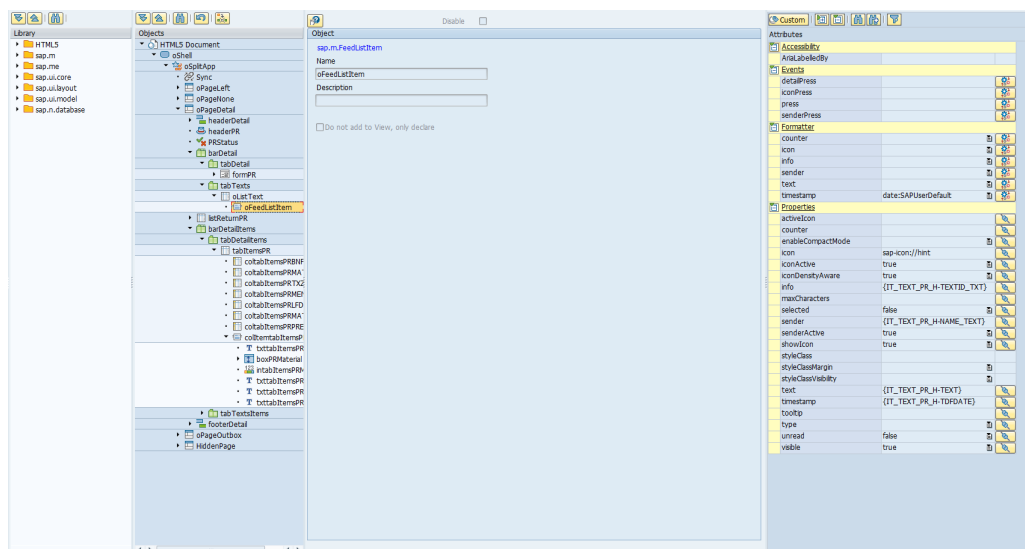
* Get Workflow items to user
CALL FUNCTION 'SAP_WAPI_CREATE_WORKLIST'
EXPORTING
  USER           = WA_HRUS_D2-US_NAME
  READ_TASK_TEXT = 'X'
  IM_TASK_FILTER = IT_TASK_FILTER
  IM_STATUS_FILTER = IT_STATUS_FILTER
TABLES
  WORKLIST       = IT_WF_LIST.

IF WA_HRUS_D2-US_NAME NE SY-UNAME.
  GV_SUBSTITUTE_USER = WA_HRUS_D2-US_NAME.
ELSE.
  GV_SUBSTITUTE_USER = SY-UNAME.
ENDIF.
...

```

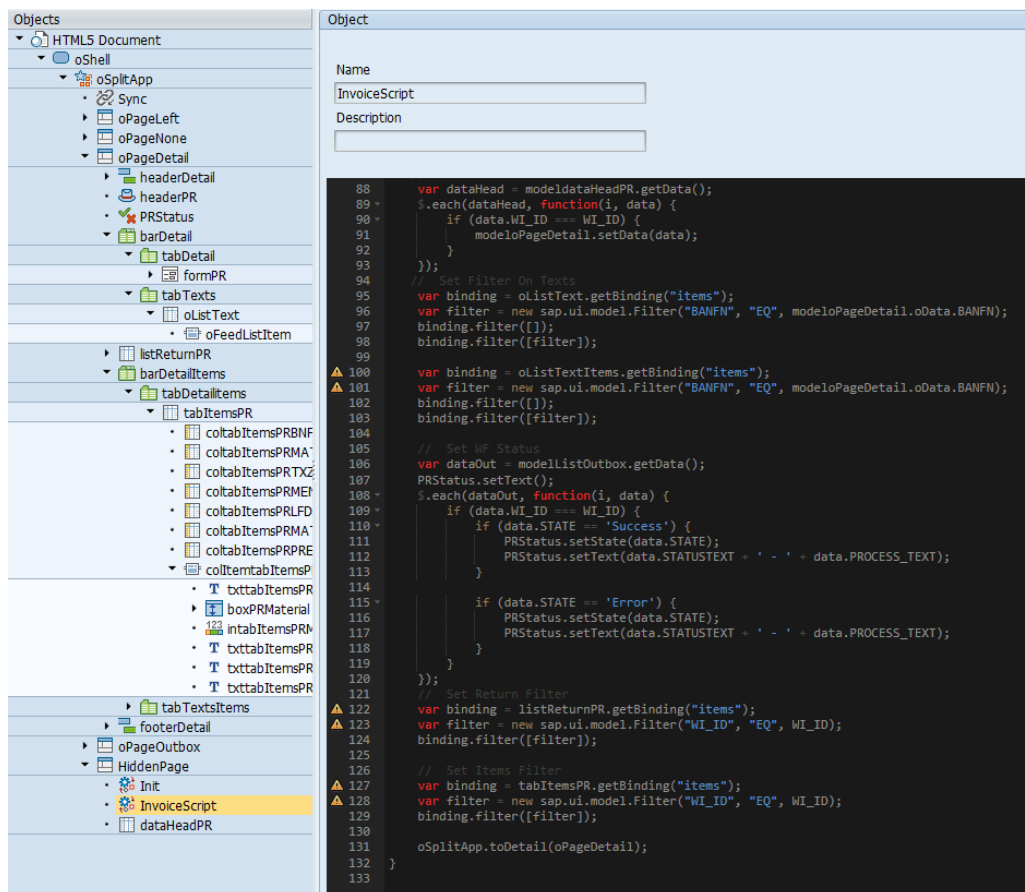
Výpis 3: Ukázka funkce v programovacím jazyce ABAP

Grafická část aplikace se skládá z komponent, které jsou součástí SAPUI5 frameworku, a postupným skládáním vzniká konečná aplikace. Samotný Neptune má propracované vývojové prostředí a většinu komponent je možné ve vývojovém prostředí přenést stylem drag & drop. Je však nutné mít dobré znalosti Javascriptu pro dopsání funkcionality aktivních prvků aplikace.



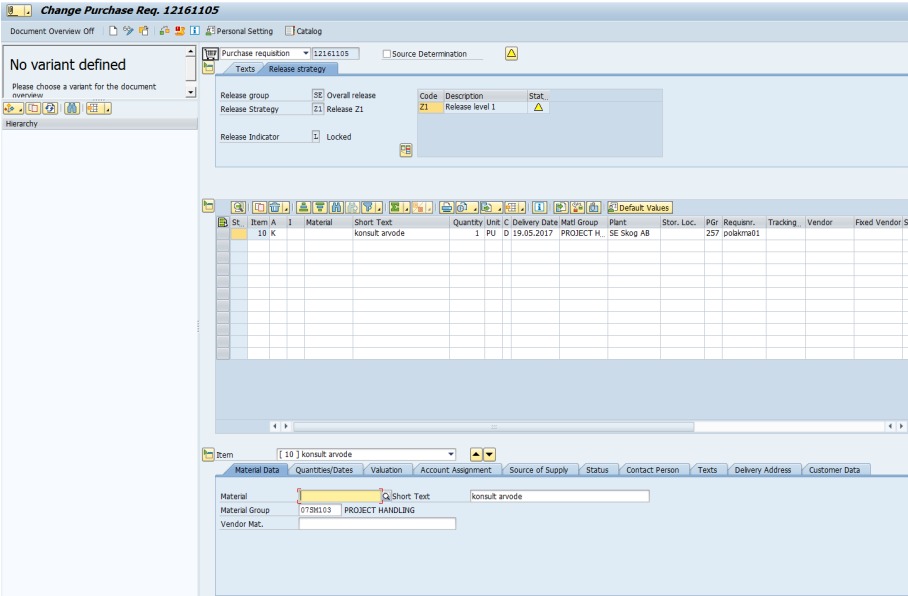
Obrázek 21: Ukázka vývoje aplikace v platformě NEPTUNE - vlastní zpracování

Na obrázku č. 22 můžeme vlevo vidět komponenty, které je možné v aplikaci použít. Vedle této části se nachází seznam použitých a poskládaných komponent, které tvoří hotovou aplikaci. V pravé části jsou zobrazeny atributy, které můžeme u vybrané komponenty změnit. U vybrané komponenty je také možnost nadefinovat události. Pro každou událost je nutné dopsat funkci, která se má vykonat.

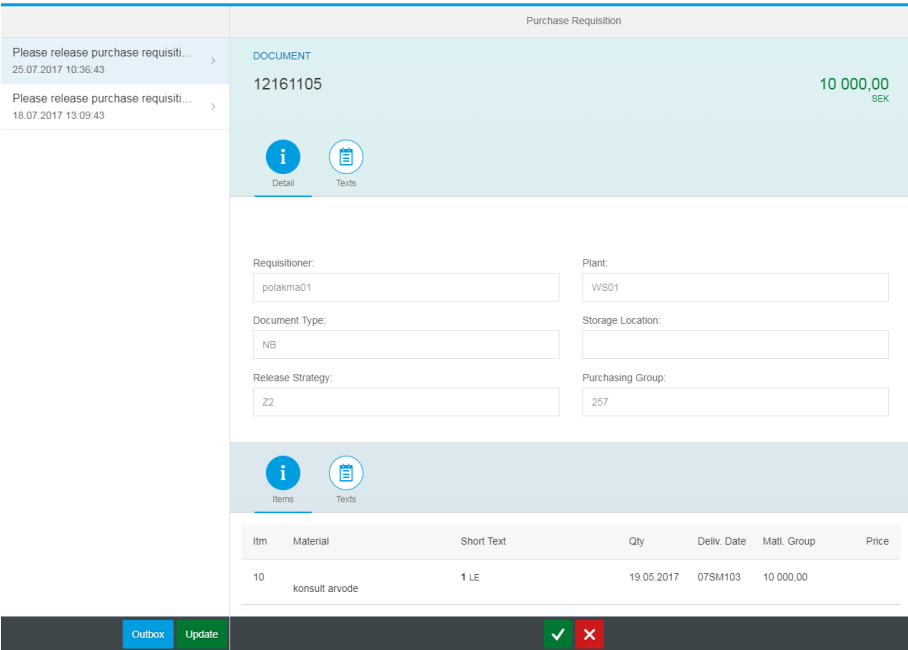


Obrázek 22: Ukázka funkce pro zobrazení stránky s podrobnými informacemi o dokumentu - vlastní zpracování

Výsledná srovnání mobilní aplikací s desktopovou aplikací SAP GUI. Výsledná aplikace na první pohled vypadá daleko jednodušeji, ale dokáže desktopovou aplikaci v rámci schvalování plně nahradit.



Obrázek 23: Původní aplikace v SAP GUI (transakční kód ME52N) - vlastní zpracování



Obrázek 24: Výsledná aplikace pro schvalování faktur na platformě Neptune - vlastní zpracování

## 5 Uplatnění teoretických a praktických znalosti a dovednosti

V průběhu své bakalářské praxe jsem využil znalosti hlavně z předmětů zaměřených na algoritmy. Z důvodu velkého množství dat v informačním systému je potřeba většinu programu dobře optimalizovat. Dále jsem zde použil práci s databází SQL pomocí jazyku OpenSQL. Tuto znalost dotazovacího jazyku jsem si nastudoval během své praxe. Při mobilním vývoji jsem hlavně využil zkušenosti z předmětu Tvorba mobilních aplikací I, a to hlavně práce s Javascriptem a práci s Adobe PhoneGap, který slouží k vytvoření instalačního balíčku. Při své práci jsem také využil znalosti z předmětu Správa operačních systému, který mi pomohl s prací se systémem Linux, na kterém firemní SAP systémy fungují.

Kromě znalosti získaných během svého studia jsem získal mnoho zkušenosti vlastní iniciativou a studiem návodů z internetu. Tyto návody se hlavně týkaly právě systému SAP, se kterým jsem neměl možnost se dříve setkat. Další zkušenosti jsem získal z doporučené literatury, kterou jsem obdržel od společnosti. Dále jsem měl možnost se účastnit kurzů na stránkách OpenSAP, případně se účastnit oficiálních školení, které se ve společnosti konaly. Během své praxe jsem se setkal s řešením, která odporují teoretickým zásadám, případně bylo nutné daná řešení navrhnout pro systémy, které už nejsou podporovány, případně pracovat s existujícími aplikacemi, ke kterým chybí jakákoliv dokumentace.

## 6 Scházející znalosti a dovednost

Jelikož jsem se nikdy dříve se systémem SAP nesetkal, byly pro mě všechny informace nové. Musel jsem se všechno teprve naučit. K tomuto účelu jsem přístup k materiálům, které slouží ke školení. Všechny vývojové aplikace se skrývají pod určitým transakčním kódem, které jsem si ze začátku nepamatoval. Postupně jsem si je osvojil tím, že jsem si kódy zaznamenával, a tímto opakováním se pro mě stávaly stále přehlednější a pro mě zapamatovatelné.

Hlavním nedostatkem byla neznalost programovacího jazyku ABAP, který jsem se musel naučit. Většinu věcí jsem se naučil ze školicích materiálu, a poté z existujících aplikací, které jsem věděl, jak použít. Při vývoji každé aplikace se dbá na maximální využití standardních funkcí kvůli budoucímu použití, které mi často ušetřily mnoho práce.

## 7 Dosažené výsledky

Mezi hlavní výsledky patří úspěšné zmigrované SAP systémy do nového datacentra. Na tomto HotCab projektu jsem měl za úkol napsat aplikace, které nám umožnily správně analyzovat data, a přizpůsobit tomu tak budoucí prostředí pro všechny uživatele, kteří se systémem pracují. Při tomto projektu jsem získal mnoho znalostí a zkušeností i z jiných odvětví, např. Networking a funkci firewallu, kde jsem měl možnost komunikovat a zlepšovat se v angličtině. Mezi další velmi přínosné zkušenosti je práce v týmu.

Během praxe jsem se naučil orientovat v cizím kódu a upravovat případné chyby. Hlavně v mobilních aplikacích, kde občas docházelo k nevysvětlitelným chybám, které byly dohledatelné jen při podrobném zkoumání. Zároveň jsem měl možnost se seznámit s Windows i Linux servery v korporátní sféře.

Zjistil jsem, že jsem se v některých věcech mýlil, a některé aplikace, které vypadaly jednoduše, byly nakonec docela složité. Poté bylo složité odhadnout časovou náročnost, kterou jsem nad danou aplikací strávil. Vždy bylo něco, co mě překvapilo a způsobilo zdržení. Jeden z příkladů může být program pro kontrolu transportů, kde z důvodu mnoha existujících objektů bylo nutné se k aplikaci vracet zpátky a snažit se eliminovat falešné poplachy, které se naskytly.

Domnívám se, že během své odborné praxe, jsem se dozvěděl mnoho nových věcí, které bych jinak neměl možnost získat. Také jsem zjistil, že komunikace s lidmi ve firmě je velmi důležitá a dokáže ušetřit spoustu času, pokud má člověk možnost využít získaných zkušeností svých starších kolegů. Rovněž obohacující byla možnost zlepšit komunikaci v anglickém jazyce.

## 8 Závěr

Když jsem do firmy nastupoval, netušil jsem, že budu moci vyhotovit bakalářskou práci formou praxe. Při této odborné praxi, která navazovala na mou brigádu, jsem získal bohaté zkušenosti se systémem SAP. Myslím si, že jako programátor jsem byl schopen většinu problémů vyřešit, a v případě problému jsem se měl vždy možnost poradit se svými spolupracovníky, kteří mi vždy ochotně poradili, a proto jsem mohl dále pokračovat v řešení. Během praxe jsem byl součástí technického týmu, který řeší i další incidenty.

Řešení těchto incidentů nebylo v mé kompetenci. Později bych však chtěl mít zkušenosti i v tomto odvětví a řešit i ostatní problémy, které se každodenně vyskytují. Při psaní této odborné práce stále pracuji ve společnosti a snažím se pomáhat při vývoji aplikací, které jsou potřebné pro implementaci dalších projektu, např. Concur a SAP SuccessFactors. Dle mého názoru je mnohem zajímavější a cennější absolvovat odbornou bakalářskou praxi místo klasické bakalářské práce. Člověk získá mnoho zkušeností v reálné korporátní sféře a zjistí, jak to ve firmě funguje. Mezi další výhodou patří, že tuto práci si už můžu zapsat do svého životopisu a může být potencionálně zajímavější pro budoucího zaměstnavatele.

Díky této práci jsem měl možnost seznámit se s novými systémy a pracovat s nimi. Také jsem měl možnost účastnit se SAP Fóra 2017 v Praze nebo navštívit Švédsko, a poznat tak další lidi, jinou kulturu a fungování společnosti v zahraničí.

## Literatura

- [1] Stora Enso Ostrava - ostravská součást světové korporace s kořeny v 13. století [online]. Magazín Patriot, 2017 [cit. 2018-04-01]. Dostupné z: <http://www.patriotmagazin.cz/stora-enso-ostrava-8211-ostravska-soucast-svetove-korporace-s-koreny-v-13-stoleti>
- [2] MAASSEN, André. SAP R/3: kompletní průvodce. Brno: Computer Press, 2007. Informační systémy. ISBN 978-80-251-1750-7.
- [3] BAVARAJU, Anil. SAP Fiori implementation and development. 2nd edition. Boston: SAP Press, Rheinwerk Publishing, 2017. ISBN 9781493215416.
- [4] VÁCLAVEK, Petr. JavaScript: hotová řešení. Brno: Computer Press, 2003. K okamžitému použití (Computer Press). ISBN 80-7226-854-6.
- [5] Neptune Software [online]. [cit. 2018-04-01]. Dostupné z: <https://template.neptune-software.com/neptune/sdk/explored>
- [6] ABAP. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-04-01]. Dostupné z: <https://en.wikipedia.org/wiki/ABAP>
- [7] SAP ERP. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-04-01]. Dostupné z: [https://en.wikipedia.org/wiki/SAP\\_\\_ERP](https://en.wikipedia.org/wiki/SAP__ERP)
- [8] SAPUI5 vs. OpenUI5. SAPUI5 SDK [online]. 2017 [cit. 2018-04-01]. Dostupné z: <https://sapui5.hana.ondemand.com/#/topic/5982a9734748474aa8d4af9c3d8f31c0.html>
- [9] SAP GUI. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-04-01]. Dostupné z: <https://en.wikipedia.org/wiki/SAPgui>